Director 8.5.1 Addendum

Introduction

Director 8.5.1 introduces two new features. The first is a new welcome movie that opens automatically the first time you launch Director 8.5.1. It contains links to four new tutorials designed to make it easier to get started using 3D and Macromedia Flash cast members. You can open the welcome movie at any time by choosing Welcome from the Director Help menu.

Director 8.5.1 also gives you the ability to use anti-aliasing with 3D cast members in your movies. Anti-aliasing improves the appearance of graphics by smoothing the lines between shapes or areas of different color so that the lines do not appear jagged. When you use anti-aliasing with a 3D sprite, the edges of each model in the sprite appear smoother against each other and against the background. Anti-aliasing of 3D sprites is particularly well-suited for merchandise demos and other e-commerce applications because its image quality is high and it can be turned on and off, as needed, in real time.

Applying anti-aliasing

To use anti-aliasing, you write Lingo scripts that enable that property for each 3D sprite you want to anti-alias.

Effects of anti-aliasing

An anti-aliased 3D sprite uses more processor power and memory than one that is not anti-aliased, resulting in lower frame rates. Because of this, it is recommended that you turn off anti-aliasing for 3D sprites while any part of the sprite is being moved or animated and turn it back on when the animation is complete. Movies designed to animate quickly, such as games, may work better with anti-aliasing turned off. During authoring, movies that use anti-aliasing continue to draw heavily on the processor, even after the movie is stopped. You may want to turn off anti-aliasing each time you stop your movie to ensure that Director's performance is not affected.

Determining whether anti-aliasing is supported

Not all 3D renderers can perform the additional calculations that anti-aliasing requires. If you have a 3D sprite that you want to anti-alias, check first that the 3D renderer supports anti-aliasing. The renderers that currently support anti-aliasing include Director's software renderer, and DirectX 5.2 and DirectX 7.0.

If the 3D sprite is in channel 1 of the Score, you would test the `antiAliasingSupported` property of sprite 1:

```
if sprite(1).antiAliasingSupported = TRUE then
```

Turning on anti-aliasing

If the `antiAliasingSupported` property is `TRUE`, you can turn on anti-aliasing for the 3D sprite by setting the sprite's `antiAliasingEnabled` property to `TRUE`.

```
sprite(1).antiAliasingEnabled = TRUE
```

For example, if you have a 3D sprite in channel 5 and you want to turn on anti-aliasing for the sprite when it first appears on the Stage, you would write a `beginSprite` script and attach it to the sprite. Your script should contain Lingo like this:

```
on beginSprite
  -- check whether anti-aliasing is supported by the current 3D renderer
  if sprite(5).antiAliasingSupported = TRUE then
    -- if it is, turn on anti-aliasing for the sprite
    sprite(5).antiAliasingEnabled = TRUE
  end if
end beginSprite
```

Turning off anti-aliasing

If you plan to animate any part of a 3D sprite, you may want to turn anti-aliasing off temporarily in order to improve the animation performance. To do this, set the `antiAliasingEnabled` property for the sprite to `FALSE`. You can then set it back to TRUE when the animation is complete.

It is a good idea to turn anti-aliasing on and off on separate handlers. For example, you might want to animate a model, camera, or light while the mouse button is held down and stop the animation when the mouse button is released. In that case you would turn off anti-aliasing in a `mouseDown` handler and turn it back on in a `mouseUp` handler.

```
on mouseDown
  -- user interaction/animation is about to start so turn
  -- anti-aliasing OFF
  sprite(1).antiAliasingEnabled = FALSE

  -- start animation
end

on mouseUp
  -- stop animation

  -- the interaction/animation has ended so turn
  -- anti-aliasing ON
  sprite(1).antiAliasingEnabled = TRUE
end
```

Lingo Dictionary

antiAliasingSupported

Syntax

```
sprite(whichSprite).antiAliasingSupported
```

Type

3D sprite property

Description

This property indicates whether anti-aliasing is supported by the current 3D renderer. This property can be tested but not set. This property returns either `TRUE` or `FALSE`.

Example

This Lingo checks whether the currently running 3D renderer for sprite 3 supports anti-aliasing. If anti-aliasing is supported, the second statement turns on anti-aliasing for the sprite with the `antiAliasingEnabled` property.

```
if sprite(3).antiAliasingSupported = TRUE then
  sprite(3).antiAliasingEnabled = TRUE
end if
```

See Also

`antiAliasingEnabled, renderer, rendererDeviceList`

antiAliasingEnabled

Syntax

`sprite(`*`whichSprite`*`).antiAliasingEnabled`

Type

3D sprite property

Description

This property indicates whether the 3D world in the sprite *`whichSprite`* is anti-aliased. It can be tested and set. The default value is `FALSE`, indicating that anti-aliasing is off. If the `antiAliasingEnabled` property is set to `TRUE` and the 3D renderer changes to a renderer that does not support anti-aliasing, the property is set to `FALSE`. The value of this property is not saved when the movie is saved.

Anti-aliased sprites use more processor power and memory than sprites that are not anti-aliased. Temporarily turning off anti-aliasing can improve the performance of animations and user interaction.

Example

This Lingo checks whether the currently running 3D renderer for sprite 2 supports anti-aliasing with the `antiAliasingSupported` property. If anti-aliasing is supported, the second statement turns on anti-aliasing for the sprite with the `antiAliasingEnabled` property.

```
if sprite(2).antiAliasingSupported = TRUE then
  sprite(2).antiAliasingEnabled = TRUE
end if
```

See Also

`antiAliasingSupported, renderer, rendererDeviceList`